



luentkkt2003 /
repo



<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights



repo / Bai kiem tra so 1 / DaoDuyLuan_2174802010599.ipynb



luentkkt2003 baitap1



78a696d · 21 minutes ago



693 lines (693 loc) · 177 KB

repo / Bai kiem tra so 1 / DaoDuyLuan_2174802010599.ipynb

↑ Top

Preview

Code

Blame

Raw



In [105...

```
import pandas as pd
import matplotlib as plt
```

In [106...

```
data = [
    ["Alice", 25, 50000],
    ["Bob", 30, 60000],
    ["Charlie", 35, 70000],
    ["David", 28, 55000],
    ["Eva", 22, 52000],
    ["Frank", 45, 80000],
    ["Grace", 34, 72000],
    ["Hannah", 31, 68000],
    ["Ivan", 27, 61000],
    ["Jack", 29, 59000],
    ["Kelly", 33, 63000],
    ["Liam", 40, 77000],
    ["Mona", 26, 53000],
    ["Nina", 32, 66000],
    ["Oscar", 36, 75000],
]

df = pd.DataFrame(data, columns=["Name", "Age", "Salary"])
print(df)
```

	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	David	28	55000
4	Eva	22	52000
5	Frank	45	80000
6	Grace	34	72000
7	Hannah	31	68000
8	Ivan	27	61000
9	Jack	29	59000
10	Kelly	33	63000
11	Liam	40	77000
12	Mona	26	53000
13	Nina	32	66000
14	Oscar	36	75000

In [107...

```
#2
df.head(15)
```

Out[107...

	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	David	28	55000
4	Eva	22	52000
5	Frank	45	80000
6	Grace	34	72000
7	Hannah	31	68000
8	Ivan	27	61000
9	Jack	29	59000
10	Kelly	33	63000
11	Liam	40	77000
12	Mona	26	53000

13 Nina 32 66000

14 Oscar 36 75000

```
In [108... df_lonhon28 = df.query("Age > 28")
```

```
In [109... print(df_lonhon28)
```

	Name	Age	Salary
1	Bob	30	60000
2	Charlie	35	70000
5	Frank	45	80000
6	Grace	34	72000
7	Hannah	31	68000
9	Jack	29	59000
10	Kelly	33	63000
11	Liam	40	77000
13	Nina	32	66000
14	Oscar	36	75000

```
In [110... #bai4
Tb_salary = df['Salary'].mean()
print(Tb_salary)
```

64066.666666666664

```
In [111... #bai5
Nhom_age = df.groupby('Age')['Salary'].sum()
print(Nhom_age)
```

Age	Salary
22	52000
25	50000
26	53000
27	61000
28	55000
29	59000
30	60000
31	68000
32	66000
33	63000
34	72000
35	70000
36	75000
40	77000
45	80000

Name: Salary, dtype: int64

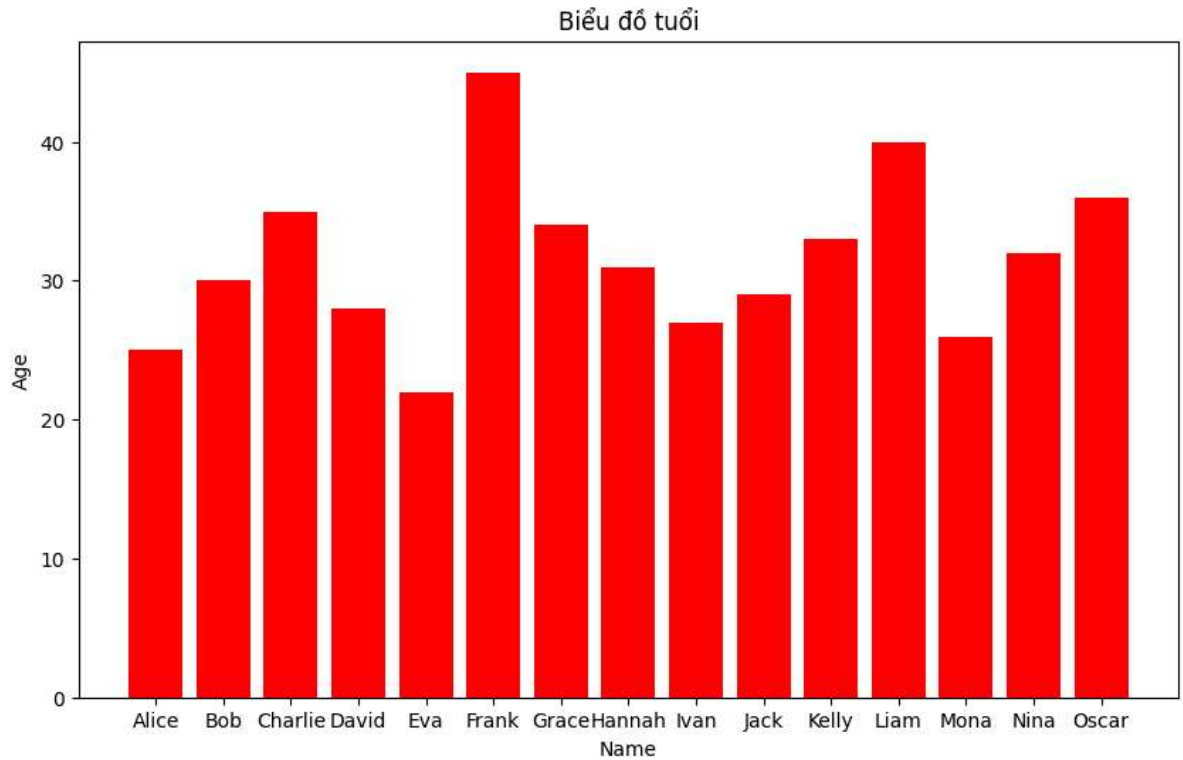
```
In [112... #bai6
df_giamdan = df.sort_values(by='Salary', ascending=False)
print(df_giamdan)
```

	Name	Age	Salary
5	Frank	45	80000
11	Liam	40	77000
14	Oscar	36	75000
6	Grace	34	72000
2	Charlie	35	70000
7	Hannah	31	68000
13	Nina	32	66000
10	Kelly	33	63000
8	Ivan	27	61000
1	Bob	30	60000
9	Jack	29	59000
3	David	28	55000
12	Mona	26	53000
4	Eva	22	52000
0	Alice	25	50000

In [113...

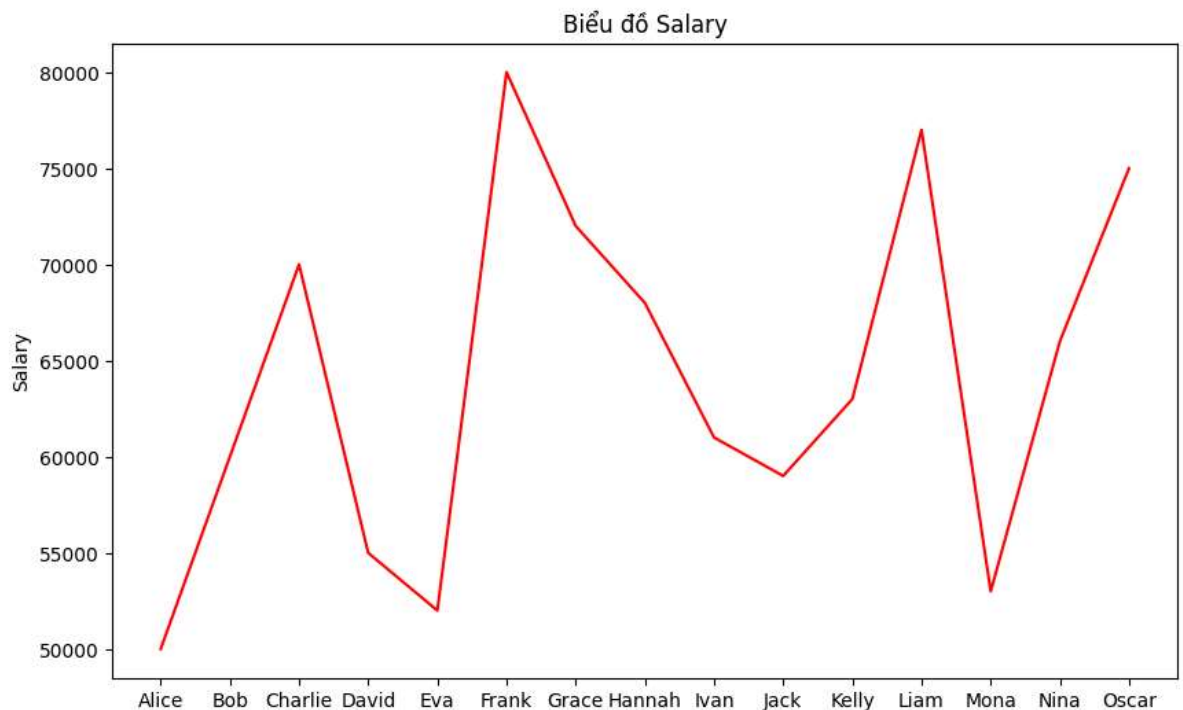
```
#Bai7
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.bar(df['Name'], df['Age'], color='red')
plt.xlabel('Name')
plt.ylabel('Age')
plt.title('Biểu đồ tuổi')

plt.show()
```



In [114...

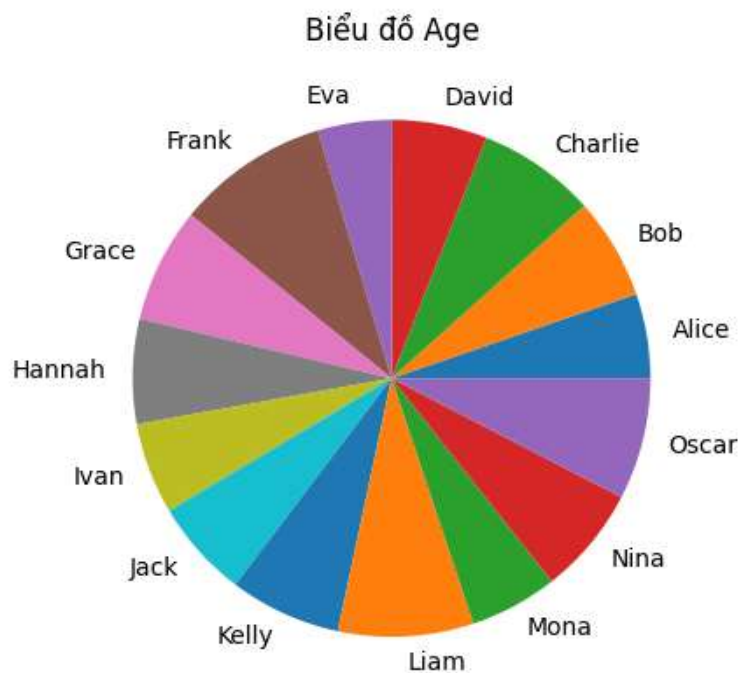
```
#bai8
plt.figure(figsize=(10, 6))
plt.plot(df['Name'], df['Salary'], color = 'red')
plt.title('Biểu đồ Salary')
plt.xlabel('Name')
plt.ylabel('Salary')
plt.show()
```



Name

In [115...

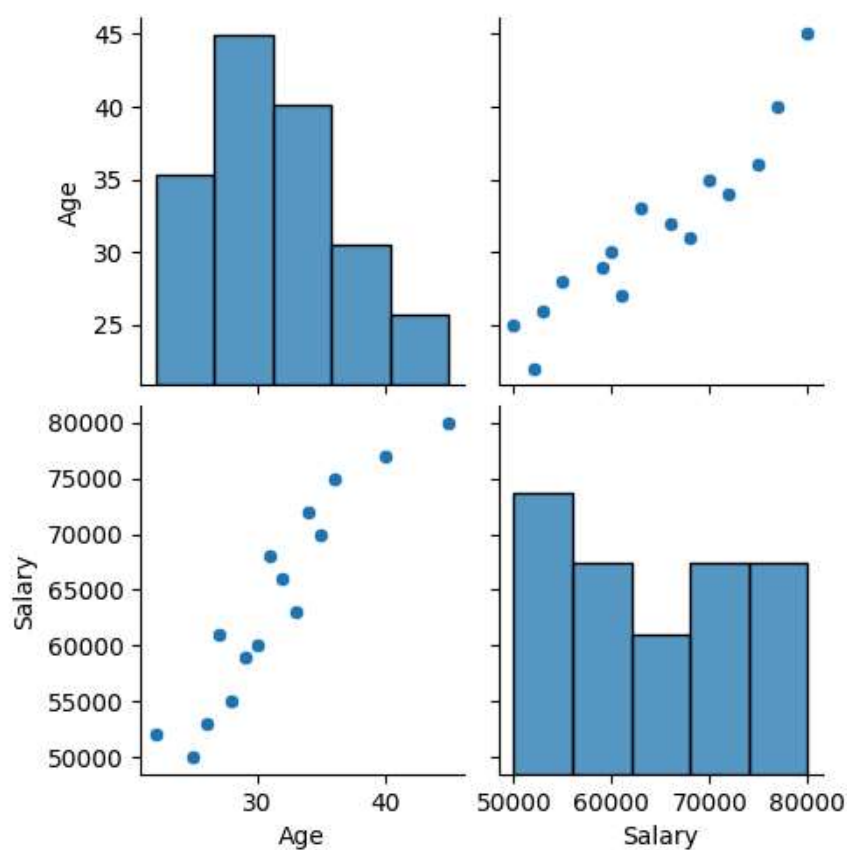
```
#bài9
plt.pie(df['Age'], labels=df['Name'])
plt.title('Biểu đồ Age')
plt.show()
```



In [116...

```
#bài10
import seaborn as sns
phantan=df[['Age','Salary']]
sns.pairplot(phantan)
```

Out[116... <seaborn.axisgrid.PairGrid at 0x1a6f2bad010>



In [117...

```
#Bai11
test= df.isna().sum()
print(test)
```

```
Name      0
Age        0
Salary     0
dtype: int64
```

In [118...

```
#Bai12
trB_age = df['Age'].mean()
df.loc[df['Age'] > 30, 'Age'] = trB_age
print(df)
```

	Name	Age	Salary
0	Alice	25.000000	50000
1	Bob	30.000000	60000
2	Charlie	31.533333	70000
3	David	28.000000	55000
4	Eva	22.000000	52000
5	Frank	31.533333	80000
6	Grace	31.533333	72000
7	Hannah	31.533333	68000
8	Ivan	27.000000	61000
9	Jack	29.000000	59000
10	Kelly	31.533333	63000
11	Liam	31.533333	77000
12	Mona	26.000000	53000
13	Nina	31.533333	66000
14	Oscar	31.533333	75000

C:\Users\PC\AppData\Local\Temp\ipykernel_7428\477502981.py:3: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a future version of pandas. Value '31.533333333333335' has dtype incompatible with int64, please explicitly cast to a compatible dtype first.

```
df.loc[df['Age'] > 30, 'Age'] = trB_age
```

In [119...

```
#bai13
df['Age_normalized'] = (df['Age'] - df['Age'].min()) / (df['Age'].max() - df['Age'].min())
print(df)
```

	Name	Age	Salary	Age_normalized
0	Alice	25.000000	50000	0.314685
1	Bob	30.000000	60000	0.839161
2	Charlie	31.533333	70000	1.000000
3	David	28.000000	55000	0.629371
4	Eva	22.000000	52000	0.000000
5	Frank	31.533333	80000	1.000000
6	Grace	31.533333	72000	1.000000
7	Hannah	31.533333	68000	1.000000
8	Ivan	27.000000	61000	0.524476
9	Jack	29.000000	59000	0.734266
10	Kelly	31.533333	63000	1.000000
11	Liam	31.533333	77000	1.000000
12	Mona	26.000000	53000	0.419580
13	Nina	31.533333	66000	1.000000
14	Oscar	31.533333	75000	1.000000

In [120...

```
#bai14
def sapxep_age(age):
    if age <= 30:
        return 'tre'
    elif 30 < age < 60:
        return 'trung nien'
    else:
        return 'gia'
df['age_group'] = df['Age'].apply(sapxep_age)
print(df)
```

	Name	Age	Salary	Age_normalized	age_group
0	Alice	25.000000	50000	0.314685	tre
1	Bob	30.000000	60000	0.839161	tre

2	Charlie	31.533333	70000	1.000000	trung nien
3	David	28.000000	55000	0.629371	tre
4	Eva	22.000000	52000	0.000000	tre
5	Frank	31.533333	80000	1.000000	trung nien
6	Grace	31.533333	72000	1.000000	trung nien
7	Hannah	31.533333	68000	1.000000	trung nien
8	Ivan	27.000000	61000	0.524476	tre
9	Jack	29.000000	59000	0.734266	tre
10	Kelly	31.533333	63000	1.000000	trung nien
11	Liam	31.533333	77000	1.000000	trung nien
12	Mona	26.000000	53000	0.419580	tre
13	Nina	31.533333	66000	1.000000	trung nien
14	Oscar	31.533333	75000	1.000000	trung nien

In [121...

```
#bài15
df['percentage'] = df['Salary'].pct_change() * 100
print(df)
```

	Name	Age	Salary	Age_normalized	age_group	percentage
0	Alice	25.000000	50000	0.314685	tre	NaN
1	Bob	30.000000	60000	0.839161	tre	20.000000
2	Charlie	31.533333	70000	1.000000	trung nien	16.666667
3	David	28.000000	55000	0.629371	tre	-21.428571
4	Eva	22.000000	52000	0.000000	tre	-5.454545
5	Frank	31.533333	80000	1.000000	trung nien	53.846154
6	Grace	31.533333	72000	1.000000	trung nien	-10.000000
7	Hannah	31.533333	68000	1.000000	trung nien	-5.555556
8	Ivan	27.000000	61000	0.524476	tre	-10.294118
9	Jack	29.000000	59000	0.734266	tre	-3.278689
10	Kelly	31.533333	63000	1.000000	trung nien	6.779661
11	Liam	31.533333	77000	1.000000	trung nien	22.222222
12	Mona	26.000000	53000	0.419580	tre	-31.168831
13	Nina	31.533333	66000	1.000000	trung nien	24.528302
14	Oscar	31.533333	75000	1.000000	trung nien	13.636364

In [122...

```
#Bài 16
df.drop_duplicates(subset=['Name', 'Age', 'Salary'])
print(df)
```

	Name	Age	Salary	Age_normalized	age_group	percentage
0	Alice	25.000000	50000	0.314685	tre	NaN
1	Bob	30.000000	60000	0.839161	tre	20.000000
2	Charlie	31.533333	70000	1.000000	trung nien	16.666667
3	David	28.000000	55000	0.629371	tre	-21.428571
4	Eva	22.000000	52000	0.000000	tre	-5.454545
5	Frank	31.533333	80000	1.000000	trung nien	53.846154
6	Grace	31.533333	72000	1.000000	trung nien	-10.000000
7	Hannah	31.533333	68000	1.000000	trung nien	-5.555556
8	Ivan	27.000000	61000	0.524476	tre	-10.294118
9	Jack	29.000000	59000	0.734266	tre	-3.278689
10	Kelly	31.533333	63000	1.000000	trung nien	6.779661
11	Liam	31.533333	77000	1.000000	trung nien	22.222222
12	Mona	26.000000	53000	0.419580	tre	-31.168831
13	Nina	31.533333	66000	1.000000	trung nien	24.528302
14	Oscar	31.533333	75000	1.000000	trung nien	13.636364

In [123...

```
#Bài17
df.to_csv('baikiemtraso1.csv', index=True)
```

